

# DirectXTK関連（頂点情報）

## ☒ DirectXTK の主要な頂点構造体一覧（VertexTypes.h）

構造体名	含まれる要素	セマンティクス	主な用途	備考
VertexPosition	位置	POSITION	最小構成（点・線など）	シェーダーで SV_Position だけ使う場合
VertexPositionColor	位置+頂点色	POSITION COLOR	頂点ごとに色を持つラインやデバッグ描画	2D線やワイヤーフレーム表示向き
VertexPositionTexture	位置+UV	POSITION TEXCOORD	テクスチャ貼り付け（Quad, Billboard, Sprite3Dなど）	☒ 最も汎用的で、現在あなたが使っている構成
VertexPositionNormal	位置+法線	POSITION NORMAL	照明（ライティング）を行う3Dメッシュ	影・陰影処理に使う基本形
VertexPositionNormalColor	位置+法線+頂点色	POSITION NORMAL COLOR	頂点色と照明を併用する場合	ゲームモデルなどでよく使われる
VertexPositionNormalTexture	位置+法線+UV	POSITION NORMAL TEXCOORD	ライティング+テクスチャ	☒ モデル描画で最も一般的
VertexPositionColorTexture	位置+色+UV	POSITION COLOR TEXCOORD	色つきUI・デカールなど	色とテクスチャを混ぜたいとき
VertexPositionNormalTangentTexture	位置+法線+接線+UV	POSITION NORMAL TANGENT TEXCOORD	法線マップ（ノーマルマップ）を使う高品質シェーディング	☼ 高級ライティング用
VertexPositionDualTexture	位置+UV×2	POSITION TEXCOORD0 TEXCOORD1	マルチテクスチャ合成	例：ライトマップ+ディフューズ

## ☒ それぞれの使い分け方（実例つき）

### 1☒ VertexPositionTexture

```
VertexPositionTexture v[] = {  
    {{ -1, 1, 0 }, { 0, 0 }},  
    {{ 1, 1, 0 }, { 1, 0 }},  
    {{ 1, -1, 0 }, { 1, 1 }},  
    {{ -1, -1, 0 }, { 0, 1 }},  
};
```

→ **Quad**や**2Dスプライト**、UI表示などに最適。

`BasicEffect` と組み合わせると、`SetTextureEnabled(true)` で完結。

## 2☒ VertexPositionNormalTexture

```
VertexPositionNormalTexture v[] = {  
    { 0,1,0}, {0,1,0}, {0,0},  
    { 1,0,0}, {0,1,0}, {1,1},  
    ...  
};
```

→ **ライトを当てたい3Dモデル**や **FBX** などを使う。

`BasicEffect::SetLightingEnabled(true)` を使えば、方向光・点光源も一瞬で追加できる。

## 3☒ VertexPositionNormalTangentTexture

```
VertexPositionNormalTangentTexture v[] = {  
    { pos, normal, tangent, uv },  
};
```

→ **ノーマルマップ対応モデル**用。

DirectXTK の `SkinnedEffect` や `NormalMapEffect` と組み合わせると最短で使える。

## 4☒ VertexPositionColor

```
VertexPositionColor v[] = {  
    { 0,0,0}, Colors::Red },  
    { 1,0,0}, Colors::Green },  
};
```

→ **デバッグ線・グリッド表示・ワイヤーフレーム**描画など。

`BasicEffect` の `SetVertexColorEnabled(true)` をオンにすれば、色付き描画可能。

# ⚙️ 裏側の仕組み（共通点）

DirectXTK の各頂点構造体は以下のような定義を持っています：

```
struct VertexPositionTexture  
{  
    DirectX::XMFLOAT3 position;  
    DirectX::XMFLOAT2 textureCoordinate;  
  
    static const D3D11_INPUT_ELEMENT_DESC InputElements[2];  
    static const UINT InputElementCount = 2;  
};
```

`InputElements` の定義はライブラリ内にあり、

`BasicEffect` や `CreateInputLayout()` にそのまま渡せます：

```
m_device->CreateInputLayout(  
    VertexPositionTexture::InputElements,  
    VertexPositionTexture::InputElementCount,  
    bytecode, bytecodeLength,  
    m_inputLayout.ReleaseAndGetAddressOf());
```

“ ☒ 自分で D3D11\_INPUT\_ELEMENT\_DESC を書かなくてよくなる！

## ☒ 他のEffectとの対応

Effectクラス	対応する頂点構造体
BasicEffect	VertexPosition, VertexPositionColor, VertexPositionTexture, VertexPositionNormalTexture
SkinnedEffect	VertexPositionNormalTexture (+ボーン情報)
NormalMapEffect	VertexPositionNormalTangentTexture
EnvironmentMapEffect	VertexPositionNormalTexture
DualTextureEffect	VertexPositionDualTexture
AlphaTestEffect	VertexPositionTexture (アルファテスト付きPS)

“ 各 Effect は自分が想定している頂点構造に合わせて内部シェーダを持っています。  
つまり、構造体を正しく選ぶだけで `InputLayout` とシェーダの整合性が取れる。

## ☒ まとめ

目的	頂点構造体	使用する Effect
シンプルなテクスチャ付き板ポリ	VertexPositionTexture	BasicEffect
光を当てたい3Dモデル	VertexPositionNormalTexture	BasicEffect
ノーマルマップ付き高品質モデル	VertexPositionNormalTangentTexture	NormalMapEffect
色付きライン・ワイヤー	VertexPositionColor	BasicEffect
2枚テクスチャ合成	VertexPositionDualTexture	DualTextureEffect

⌚ Revision #2

★ Created 27 October 2025 06:21:18 by youe2

✍ Updated 9 June 2026 00:16:43 by youe2